Amendments to the Claims

1. (Currently Amended) A classfile modification method, comprising:

modifying a classfile after said classfile has been compiled from source code, said classfile describing properties of a class within an object oriented environment, said modifying comprising:

modifying a method information structure by adding byte code instructions to the byte code instructions of said method information structure's respective method, said byte code instructions to cause a plug-in module's handler method to execute an output function for said method; ~~and,~~

adding a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions[[.]];

compiling results of the modifying of the classfile, the results including method information; and

filtering the method information by applying filtering parameters via a filtering module, the filtering of the method information including filtering timing data, method invocations, and other method-related information.

2. (Original) The classfile modification method of claim 1 wherein said

identities are each in a character string format.

3. (Original) The classfile modification method of claim 2 wherein said modifying a classfile further comprises:

adding a field information structure, said field information structure describing a field that is to store a numeric identifier of said class.

4. (Original) The classfile modification method of claim 3 wherein said numeric identifier is provided to said class by a method of which said dispatch unit is comprised.

5. (Original) The classfile modification method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an entry point of said method being reached.

6. (Original) The classfile modification method of claim 5 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method;

2) recording an input parameter value for said method; and,

3) incrementing a counter for said method.

7. (Original) The classfile modification method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an exit point of said method being inevitably reached.

8. (Original) The classfile modification method of claim 7 wherein said

output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method;
2) recording an input parameter value for said method; and,
3) incrementing a counter for said method.

9. (Original) The classfile modification method of claim 7 wherein portions of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to any exit point of said method being inevitably reached.

10. (Original) The classfile modification method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising during execution of said method.

11. (Original) The classfile modification method of claim 1 wherein:

a first portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute said output function treatment in response to an entry point of said method being reached;

a second portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute

said output function treatment in response to an exit point of said method being inevitably reached; and,

a third portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute said output function treatment in response to an error arising during execution of said method.

12 (Original) The classfile modification method of claim 1 wherein at least one of said instructions invokes a second method of which said dispatch unit is comprised.

13 (Original) The classfile modification method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokestatic instruction.

14. (Original) The classfile modification method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokevirtual instruction.

15. (Original) The classfile modification method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokespecial instruction.

16. (Original) The classfile modification method of claim 12 wherein said second method references a dictionary that correlates a numeric identification of said method and said class to a location where said plug-in module is found.

17. (Original) The classfile modification method of claim 1 wherein said modifying of said classfile further comprises modifying a second method information structure by adding byte code instructions to said second method information structure's respective method, said byte code instructions to cause a second plug-in module's handler to execute output function treatment for said respective method.

18. (Original) The classfile modification method of claim 17 wherein said second method is a constructor.

19. (Original) The classfile modification method of claim 1 further comprising adding byte code level instructions that assign numeric names to said classfile's methods in lieu of character string names.

20. (Original) The classfile modification method of claim 19 wherein said numeric names are based upon the order in which said methods are listed in said classfile, each next method in said order having a numeric name equal to a fixed increment above the numeric name for its immediately preceding method in said order.

21. (Original) The classfile modification method of claim 20 wherein said byte code instructions for registering are configured to execute in response to said classfile being loaded.

22. (Currently Amended) A machine readable storage medium containing instructions which when executed cause a classfile modification method to be performed, said ~~byte code~~ classfile modification method comprising:

modifying a classfile after said classfile has been compiled from source code, said classfile describing properties of a class within an object oriented environment, said modifying comprising:

modifying a method information structure by adding byte code instructions to the byte code instructions of said method information structure's respective method, said byte code instructions to cause a plug-in module's handler method to execute an output function for said method; ~~and,~~

adding a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions[[.]];

compiling results of the modifying of the classfile, the results including method information; and

filtering the method information by applying filtering parameters via a filtering module, the filtering of the method information including filtering timing data, method invocations, and other method-related information,

23. (Currently Amended) The machine readable storage medium of claim 22 wherein said identities are each in a character string format.

24. (Currently Amended) The machine readable storage medium of claim 23 wherein said modifying a classfile further comprises:

adding a field information structure, said field information structure describing a field that is to store a numeric identifier of said class.

25. (Currently Amended) The machine readable storage medium of claim 24 wherein said numeric identifier is provided to said class by a method of which said dispatch unit is comprised.

26. (Currently Amended) The machine readable storage medium of claim 22 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an entry point of said

method being reached.

27. (Currently Amended) The machine readable storage medium of claim 26 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method;

2) recording an input parameter value for said method; and,

3) incrementing a counter for said method.

28. (Currently Amended) The machine readable storage medium of claim 22 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an exit point of said method being inevitably reached.

29. (Currently Amended) The machine readable storage medium of claim 28 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method;

2) recording an input parameter value for said method; and,

3) incrementing a counter for said method.

30. (Currently Amended) The machine readable storage medium of claim 28 wherein portions of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to any exit point of said method being inevitably reached.

31. (Currently Amended) The machine readable storage medium of claim 22 wherein a portion of said byte code instructions that are added to

said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising during execution of said method.

32. (Currently Amended) The machine readable storage medium of claim 22 wherein:

a first portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute said output function treatment in response to an entry point of said method being reached;

a second portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute said output function treatment in response to an exit point of said method being inevitably reached; and,

a third portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to execute said output function treatment in response to an error arising during execution of said method.

33. (Currently Amended) The machine readable storage medium of claim 22 wherein at least one of said instructions invokes a second method of which said dispatch unit is comprised.

34. (Currently Amended) The machine readable storage medium of claim 33 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokestatic instruction.

35. (Currently Amended) The machine readable storage medium of claim 33 wherein said byte code instructions are Java compatible and wherein said

at least one of said instructions is an invokevirtual instruction.

36. (Currently Amended) The machine readable storage medium of claim 33 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invokespecial instruction.

37. (Currently Amended) The machine readable storage medium of claim 33 wherein said second method references a dictionary that correlates a numeric identification of said method and said class to a location where said plug-in module is found.

38. (Currently Amended) The machine readable storage medium of claim 22 wherein said modifying of said classfile further comprises modifying a second method information structure by adding byte code instructions to said second method information structure's respective method, said byte code instructions to cause a second plug-in module's handler to execute output function treatment for said respective method.

39. (Currently Amended) The machine readable storage medium of claim 38 wherein said second method is a constructor.

40. (Currently Amended) The machine readable storage medium of claim 22 wherein said classfile modification method further comprises adding byte code level instructions that assign numeric names to said classfile's methods in lieu of character string names.

41. (Currently Amended) The machine readable storage medium of claim 40 wherein said numeric names are based upon the order in which said methods are listed in said classfile, each next method in said order having a numeric name equal to a fixed increment above the numeric name for its immediately preceding method in said order.

42. (Currently Amended) The machine readable storage medium of claim 41 wherein said byte code instructions for registering are configured to execute in response to said classfile being loaded.

43. (New) A system comprising:

a classfile modification system having a processor and a storage medium coupled with the processor, the classfile modification system to perform a classfile modification method, said classfile modification system to:

modify a classfile after said classfile has been compiled from source code, said classfile describing properties of a class within an object oriented environment, said modifying comprising:

modify a method information structure by adding byte code instructions to the byte code instructions of said method information structure's respective method, said byte code instructions to cause a plug-in module's handler method to execute an output function for said method;

add a method information structure that includes byte code instructions for registering the identities of said class and said method with a dispatch unit that is responsible for dispatching an invocation to said plug-in module during runtime execution of said modified byte code, said invocation directed to said dispatch unit from said added byte code instructions;

compile results of the modifying of the classfile, the results including method information; and

filter the method information by applying filtering parameters via a filtering module, the filtering of the method information including filtering

timing data, method invocations, and other method-related information.

44. (New) The system of claim 43 wherein said identities are each in a character string format.

45. (New) The system of claim 44 wherein said modifying a classfile further comprises:

adding a field information structure, said field information structure describing a field that is to store a numeric identifier of said class.

46. (New) The system of claim 45 wherein said numeric identifier is provided to said class by a method of which said dispatch unit is comprised.

47. (New) The system of claim 43 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an entry point of said method being reached.